Bridging gaps in app localization
between technologists, linguists and project managers

Afaf Steiert is president and founder of Afaf Translations, works as an Arabic conference interpreter and oversees all medical translation services at Afaf Translations.

Afaf Steiert

Yasin Steiert works as an English editor with Afaf Translations. He speaks fluent Arabic and English and has a degree in international relations.

Yasin Steiert

*T*he computer-human relationship has been integrating since technology systems became available to the everyday consumer. Computers have shifted from word processing and simple computation on our desks to fitting into our pockets and providing us an assortment of tools.

The demand for more and more of these tools has resulted in a highly organic app market that retains a relatively low barrier of entry. This demand shows no sign of slowing; smartphones surpassed desktop computers in 2014, and the market is predicted to continue experiencing an exponential increase into 2020 and beyond. It is in any app developer's interest to diversify the penetrability of their product into the global markets.

To execute a successful internationalization project on an app, it is important that language service providers (LSPs) and app developers increasingly become symbiotic in their knowledge of one another. Translators have often come to us to express their discontent with projects that are related to mobile apps, since there is a disconnect between what an LSP can do and what an app developer demands for their project. Creating a clear and defined scope of the viability of the project's success and utilizing a translation management system (TMS) that the client can also be involved in will greatly assist in reducing wasted time and money for all parties involved.

The demand

LSPs that aren't attempting to educate their translators and adapt their pricing models for mobile applications are unfortunately going to miss out on large market share. Consider the distributed location of developers across the globe as shown in Figure 1. Nearly 90% of developer focus exists within the already developed or developing markets of Europe, North America and Asia. This concentration of developers is also reflective of the concentration of mobile app users and appropriately reflects the cross potential that exists in these markets. Furthermore, for a US-based LSP, it is especially fortuitous that the United States has typically shared domination of the global publishers market with Japan, as seen in Figure 2. With app developers protecting their profit margins it's only

Figure 1: Total number of mobile app developers by region. Source: Dogtiev, 2016.

natural that they will follow growth in the market (Asia, Africa and the Middle East) and seek localization services accordingly. It's been commonly understood that internationalized application will naturally have a higher rate of daily active users. That is why all LSP project managers must have a fundamental understanding of what is involved in translating an app, and they must understand the technical jargon that app developers use while conversely preparing developers for the work that localization demands from them.

App localization is far more than merely taking the text from an app and translating it. LSPs must set a strategy with the client to develop an understanding of what markets they expect to succeed in. If the client is keen on translating into one language (such as Arabic), it is much more viable to take a deep localization approach to fully integrate into a few markets. The broader the internationalization of the project, the more work there will be on the user interface (UI). Internationalization within the mobile apps setting refers to the ability of the app to adapt to different languages and present itself as a native app to the user. In order to create multiple languages for an application, translators will need to identify the UI strings from the app code in order to compile that text into multiple resource files. We often create a custom XML file to house these UI strings; the system you use is highly dependent upon your translator's capability and client's demands. A fantastic way to simplify this process across large-scale projects or highly layered and complex applications such as social media apps can be achieved by using a TMS. Large scale LSPs often develop their own, but smaller LSPs can easily remain just as competitive by utilizing various third-party TMSs offered by companies such as OneSky, Smartling or Wordbee to allow managers to monitor the
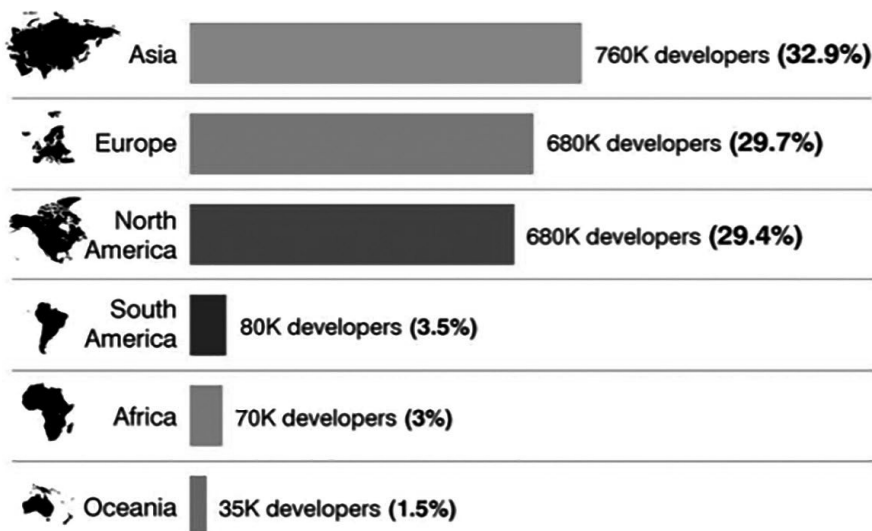
overall progress of the project. It is fundamental to include the client in your TMS to maximize the efficiency of the project.

In projects where we have worked in conjunction with other companies, we often saw an under-utilization of TMSs. Time is repetitively wasted in the process of allowing each contractor team to operate independently under the direction of separate project managers that continue to fragment client demands.

A lag would often exist between understanding the issues in the UI and how linguists could aim at solving them through the TMS. This stovepiping of information is a classic attempt at disallowing any one individual from having too much information on the project, but this highly compromises the speed and quality of the work. It perpetuates a cycle in which contractors are constantly asked to repair work they have just finished according to previously-unrefined guidelines. This compounds into contractor fatigue, which further hinders the project's timely success.
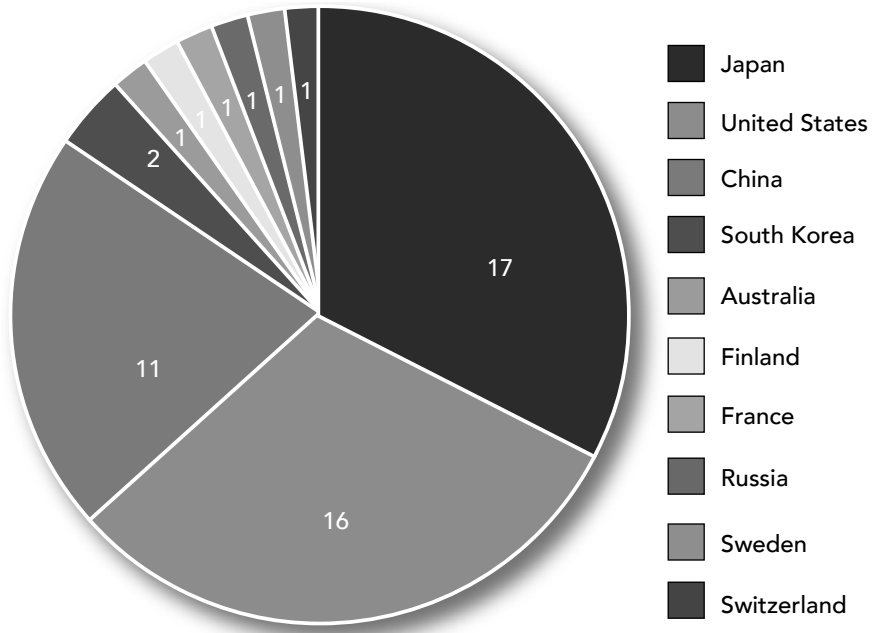
Figure 2: Top 52 publishers of 2016. Source: App Annie, 2017.

Rather than using this style of management, it is far wiser to invest your project managers' time into sourcing trustworthy contractors who can work in-house or who can be held highly accountable throughout all stages of the project. They should then be allowed more access to the project. By including them directly with the UI and allowing the client to have plenty of time to check in on how the translation impacts the app, there is less likelihood for a gap to exist between translator and developer. Developers often have a misconception that non-developers will not understand their demands, but you do not need to be a developer to understand the basic demands of a UI. It is up to the LSP to create a dialogue that can be transparently transmitted from developer to contractor.

Additionally, incorporate a localization tester who is fluent in both the source language and the target language so that they are able to view both apps. Depending upon the demands of internationalization, merely mirroring the UI and changing the text direction won't suffice for certain languages and further development will be demanded from the client. If you have these capabilities as an LSP, then your eye for detail must be prudent since internationalization demands the app to adapt to the other language, do not wait for issues to arise to provide such information. If all parties are aware of the complexity that exists in translating a mobile application and the capabilities that all parties have, then common inefficiencies can be aptly mitigated.

Finally, your project is not over when your translation is complete. Even with your own tester, components may be missed, a new demographic may begin downloading the application, or a variety of changes in the market will cause the localization project to fall short. It is a necessity for the client and LSP to continually check the app's ratings and comments in order to catch any further glitches and mistakes early before the market decidedly retires the app. If each step of this process is followed, then you shouldn't have any gaps existing between developer expectations and linguist's abilities.

The incorporation of artificial intelligence

All of this relates primarily to your traditional mobile phone app that requires analog user interaction to control it. However, a common trend in application technology had been the incorporation of artificial intelligence (AI) assistants such as Samsung's Bixby, Google Assistant or Siri. Linguists are a large part of these projects because these AI assistants must first be taught a language.

As with any specialization, the environment between developers and linguists will likely improve with time. The better an L SP can familiarize itself with the basics of app development, the more easily it can guide the client. In many client interactions there is an apparent assumption that translating language for an app should be as easy as translating a document. This is far from the truth, since unlike a document in which only one variable (words) exists, an app involves a far larger degree of complexity. In an industry that is based on clicks per minute, downloads per minute and megabits per second, it's important for LSPs to adjust developers' working culture to the language industry. We're an industry that is based on human capital, not code, and it's our job to make developers understand that. **[M]**